

Tilburg University

Linear stationary point problems on unbounded polyhedra

Talman, A.J.J.; Dai, Y.

Published in:
Mathematics of Operations Research

Publication date:
1993

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):
Talman, A. J. J., & Dai, Y. (1993). Linear stationary point problems on unbounded polyhedra. *Mathematics of Operations Research*, 18(3), 635-644.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LINEAR STATIONARY POINT PROBLEMS ON UNBOUNDED POLYHEDRA

YANG DAI AND DOLF TALMAN

In this paper we propose a complementary pivoting algorithm for finding a stationary point of an affine function on an unbounded polyhedron. Under some mild conditions there is a piecewise linear path from an arbitrarily chosen point in the polyhedron leading to a solution of the problem. By exploiting fully the linearity of the problem, each linear piece of the path is followed in principle by making just one linear programming pivoting step.

1. Introduction. Given a polyhedron K in the n -dimensional Euclidian space R^n and an affine function f from K into R^n , we consider the problem of finding a point x^* in K such that

$$x^T f(x^*) \leq (x^*)^T f(x^*) \quad \text{for all } x \in K.$$

This problem is called the linear stationary point problem and the point x^* is called a stationary point of f on K . Yamamoto (1987) proposed a path following algorithm for solving the linear stationary point problem on a polytope, i.e., on a bounded polyhedron. The algorithm starts with an arbitrary point in the polytope and generates a piecewise linear path which leads to a stationary point. The Dantzig-Wolfe decomposition method for linear programming problems is used for tracing the path. In Dai et al. (1991) the basic idea of path-following was introduced for solving the linear stationary problem on an unbounded, pointed polyhedron. However, the decomposition technique used in these algorithms may result in unnecessary linear programming pivoting steps for following a linear piece of the path. To avoid the inefficiency, Kamiya and Talman (1990) recently proposed a new algorithm for solving the linear stationary point problem on a polytope. In their algorithm each linear piece of the path is followed by making, in principle, just one linear programming pivoting step.

The main purpose of this paper is to generalize their algorithm to the linear stationary point problem on an unbounded polyhedron. This problem is frequently met in mathematical programming, for example, finding a Karush-Kuhn-Tucker point for a quadratic programming problem with linear constraints.

The organization of this paper is as follows. In §2 we show the existence of the path of the algorithm. Section 3 describes the steps of the algorithm.

2. The path of the algorithm. Throughout this paper we assume that the polyhedron K is unbounded and pointed, i.e., K has at least one vertex, and that K is represented by the set $\{x \in R^n | A^T x \leq b\}$, where A is an $n \times m$ -matrix and b an m -vector. Moreover we assume that K is simple, i.e., each vertex of K lies on n edges. Without loss of generality we assume that none of the constraints in determin-

Received October 13, 1990; revised March 19, 1992.

AMS 1980 subject classification. Primary: 90C30; Secondary: 90C33.

LAOR 1973 subject classification. Main: Programming: Nonlinear.

OR/MS Index 1978 subject classification. Primary: 653 Programming/Nonlinear; Secondary: 622 Programming/Complementarity.

Key words. Complementary pivoting, stationary point problem, linearity.

ing K is redundant and that $\dim(K) = n$. Let f be an affine function from K into R^n , i.e., $f(x) = Cx + d$ for some $n \times n$ -matrix C and n -vector d . Let w be an arbitrary point in K . In the algorithm proposed below for solving the stationary point problem on K the point w will be the starting point. For some strictly positive m -vector γ let $a^0 = -A\gamma$ and let $H^0 = \{x | (a^0)^\top x = b_0\}$ be a hyperplane for some positive b_0 . We can see that if b_0 is sufficiently large this hyperplane intersects every unbounded face of K , while the negative halfspace $H^- = \{x | (a^0)^\top x \leq b_0\}$ contains in its interior all vertices of K as well as w and hence also all bounded faces of K .

We denote the i th n -dimensional column vector of A by a^i , $i = 1, \dots, m$. Let $H^+ = \{x | (a^0)^\top x \geq b_0\}$ be the positive halfspace of H^0 . Then for a given subset $I \subset \{0, 1, \dots, m\}$ let the set F_I be defined by

$$F_I = H^- \cap \{x \in K | (a^i)^\top x = b_i \text{ for } i \in I\},$$

and when $0 \in I$ let the set F_I^+ be defined by

$$F_I^+ = H^+ \cap \{x \in K | (a^i)^\top x = b_i \text{ for } i \in I \setminus \{0\}\}.$$

Remark that $F_I = F_{I \setminus \{0\}} \cap H^0$ and $F_I = F_I^+ \cap H^0$ if $0 \in I$. An example is illustrated in Figure 1.

The normal cone of F_I for $I \subset \{0, 1, \dots, m\}$, denoted by $N(F_I)$, is defined by

$$N(F_I) = \left\{ y \in R^n | y = \sum_{i \in I} \mu_i a^i, \mu_i \geq 0 \text{ for } i \in I \right\}.$$

The following lemma describes the characterization of a solution to the linear stationary point problem (LSPP) on K .

LEMMA 2.1. *The point x in K is a stationary point of f on K if and only if for some $I \subset \{0, 1, \dots, m\}$ both $x \in F_I$ and $f(x) \in N(F_I)$ when $0 \notin I$ or both $x \in F_I^+$ and $f(x) \in N(F_{I \setminus \{0\}})$ when $0 \in I$.*

To find a stationary point of f on K , the algorithm will follow, starting at $x = w$, a piecewise linear path S in K such that for each point x on the path S one of the

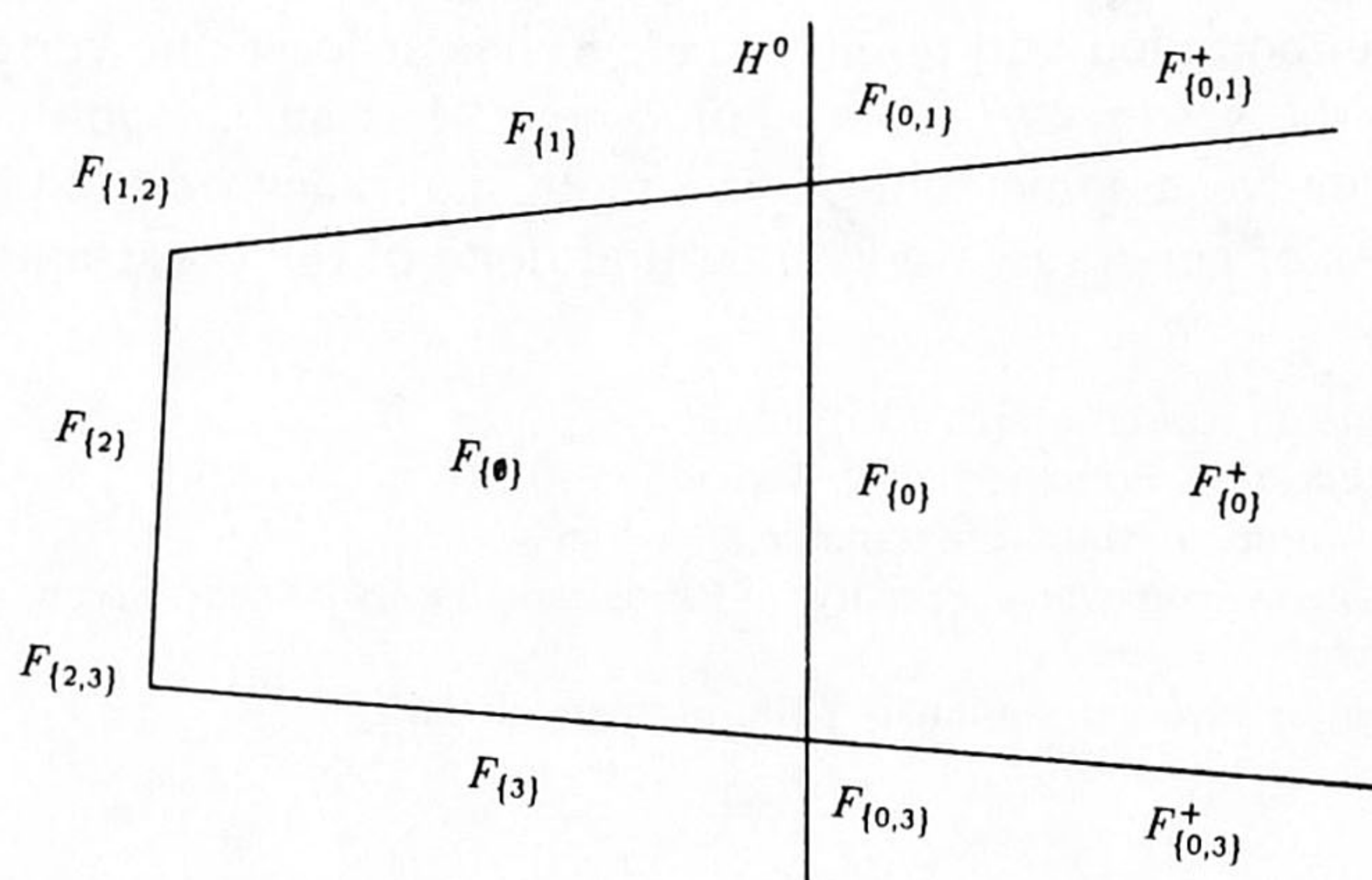


FIGURE 1.

following two cases holds for some $I \subset \{0, 1, \dots, m\}$:

- (1.i) $x = (1 - \lambda)w + \lambda z$ for some $\lambda \in [0, 1]$ and some $z \in F_I$,
- (ii) $f(x) \in N(F_I)$;
- (2.i) $x \in F_I^+$,
- (ii) $f(x) \in N(F_I)$.

Clearly, in case (1) if $\lambda = 1$ and $0 \notin I$ or in case (2) if $f(x) \in N(F_{I \setminus \{0\}})$ then x is a stationary point of f on K . The starting point $x = w$ satisfies case (1) for $\lambda = 0$ and F_I corresponding to a vertex of $K \cap H^-$. Assuming nondegeneracy, for a given index set $I \subset \{0, 1, \dots, m\}$ the points, if any, satisfying either (1.i) and (1.ii) or (2.i) and (2.ii) form a line segment. In the next section we describe how such a line segment can be followed by making just one pivot step. As in [2] these line segments can be linked for different index sets I to form piecewise linear loops and paths, one path being the path S having w as an end point. All other end points of these paths are stationary points of f on K .

Since the polyhedron K is unbounded, the path S may diverge to infinity. We need some condition on the function f under which the path S is bounded and therefore leads from the starting point w to a stationary point. Like for other complementary pivoting algorithms for solving linear complementarity problems it can be shown that if the matrix C is copositive plus on a polyhedral cone consisting of the directions of all rays in K and if the problem has a solution, then the path S cannot diverge to infinity and consequently leads to one of the stationary points of f on K . See Dai et al. (1991) for details.

3. Following the path. In this section we describe the steps of the algorithm to follow the path S in such a way that each linear piece of S is followed in just one linear programming pivoting step.

To initiate the algorithm for following the path S from $x = w$, we first need to determine the vertex w^1 of the set $K \cap H^-$ for which $f(w) \in N(\{w^1\})$. To do so we solve a linear programming problem given by

$$\begin{aligned} \text{(Primal)} \quad & \max. \quad (Cw + d)^T x \\ & \text{s.t.} \quad A^T x \leq b \\ & \quad (a^o)^T x \leq b_o, \end{aligned}$$

or

$$\begin{aligned} \text{(Dual)} \quad & \min. \quad b^T \mu + b_o \mu_o \\ & \text{s.t.} \quad A\mu + \mu_o a^o = Cw + d \\ & \quad \mu \geq 0, \mu_o \geq 0. \end{aligned}$$

Solving the Primal yields a vertex w^1 of $K \cap H^-$. From the Dual it immediately follows that $f(w) \in N(\{w^1\})$. From Lemma 2.1 we obtain that if $w = w^1$ then w is a stationary point of f on K . Note that w^1 does not lie in H^o when $w = w^1$. Suppose that $w \neq w^1$. Then the first linear piece of the path S lies on the line segment between w and w^1 . Let the index set I be defined by

$$I = \{i \mid (a^i)^T w^1 = b_i \text{ for } i \in \{0, 1, \dots, m\}\}.$$

The set I is equal to the set $\{i \mid \mu_i^o > 0 \text{ for } i \in \{0, 1, \dots, m\}\}$, where μ_i^o is the optimal value of μ_i in the Dual. Assuming nondegeneracy, the set I consists of n elements. Let D^{-1} be the inverse of the basis matrix at the solution of the Dual, i.e.,

$D = (a^i, i \in I)$. In order to follow the first linear piece of the path S from w , we make a linear programming (l.p.) pivoting step in the Dual with the vector $C(w^1 - w)$ having incoming variable λ_1 as follows. From $r = D^{-1}(C(w^1 - w))$ we obtain for $i \in I$ the direction $\bar{\mu}_i$ into which μ_i moves from μ_i^0 when λ_1 is increased from 0. Suppose that λ_1 becomes 1 before some of the μ_i 's becomes negative, then w^1 is a stationary point of f on $K \cap H^-$. If $0 \notin I$ in this case, then w^1 is a stationary point of f on K . Otherwise the second linear piece of the path S lies in the set F_I^+ . Now we need a point w^0 to determine the direction into which the algorithm will move next. We take $w^0 = w^1 + (D^\top)^{-1}e(\tilde{h})$, where \tilde{h} is the index of the row $(a^0)^\top$ in D^\top and $e(\tilde{h})$ is the \tilde{h} th unit vector in R^n . So w^0 is a point in R^n such that $(a^i)^\top w^0 = b_i$ for all $i \in I \setminus \{0\}$ and $(a^0)^\top w^0 > b_0$. The points w^0 and w^1 are such that their affine hull contains F_I^+ . From $w^1 - w^0$ we can derive the direction into which the algorithm moves next as will be described in subsection 3.2 for the general case.

Suppose now that $\mu_{i'}$ becomes first 0 before λ_1 becomes 1, for some $i' \in I$. If $w \in F_{I \setminus \{i'\}}$ and $0 \notin I \setminus \{i'\}$, then $x = (1 - \lambda_1)w + \lambda_1 w^1$ lies in $F_{I \setminus \{i'\}}$ and $f(x) \in N(F_{I \setminus \{i'\}})$, and x is a stationary point of f on K . Otherwise, the second linear piece of the path S lies in the convex hull of w and the face $F_{I \setminus \{i'\}}$ of $K \cap H^-$, where $(a^{i'})^\top x$ is allowed to become less than $b_{i'}$. To determine in which direction the algorithm will move next we need a point w^2 such that $(a^i)^\top w^2 = b_i$ for all $i \in I \setminus \{i'\}$ and $(a^{i'})^\top w^2 < b_{i'}$ for $i = i'$, for example, take $w^2 = w^1 - (D^\top)^{-1}e(\tilde{h})$, where \tilde{h} is the index of the row $(a^{i'})^\top$ in D^\top and $e(\tilde{h})$ is again the \tilde{h} th unit vector in R^n . The points w^1 and w^2 are such that their affine hull contains $F_{I \setminus \{i'\}}$. Note that the dimension of $F_{I \setminus \{i'\}}$ is equal to one. Now the index set I becomes equal to $I \setminus \{i'\}$ and we make a pivoting step in the matrix D^{-1} with $C(w^1 - w)$ in order to replace $a^{i'}$. This concludes the initialization of the algorithm.

In general the algorithm follows the subsequent linear pieces of the path S by making a sequence of l.p. pivoting steps. A linear piece of S consists of points x such that for some I either case (1) x lies in $wF_I = \{x \in R^n | x = (1 - \lambda_1)w + \lambda_1 z, z \in F_I, 0 \leq \lambda_1 \leq 1\}$ for some face F_I of $K \cap H^-$ and $f(x) \in N(F_I)$, or case (2) x lies in F_I^+ for some face F_I^+ of $K \cap H^+$ and $f(x) \in N(F_I)$. Let $\text{Aff}(X)$ denote the affine hull of a set X . In case (1), let k be such that $\dim(wF_I) = k$; then we represent F_I through k affinely independent points w^1, \dots, w^k in R^n such that

$$\text{Aff}(\{w^1, \dots, w^k\}) = \text{Aff}(F_I).$$

In case (2), let k be such that $\dim(F_I^+) = k$. Then we represent F_I^+ through $k + 1$ affinely independent points w^0, w^1, \dots, w^k in R^n such that

- (1) $\text{Aff}\{w^1, \dots, w^k\} = \text{Aff}(F_I)$;
- (2) $\text{Aff}\{w^0, w^1, \dots, w^k\} = \text{Aff}(F_I^+)$.

This representation of F_I or F_I^+ by affinely independent points allows us to follow a linear piece of the path S in wF_I or F_I^+ by making just one l.p. pivoting step. This avoids the subsequent steps of column generations for following one linear segment as done in Yamamoto (1987).

3.1. Tracing the path in wF_I . Suppose that after the initiation the algorithm is moving for some $I \subset \{0, 1, \dots, m\}$ in the convex hull of w and F_I . Let $|I| = n - k + 1$, so that $\dim(wF_I) = k$. Let w^1, \dots, w^k be affinely independent points in the affine hull of F_I . Then a point x on the path S in wF_I can be written as

- (i) $x = (1 - \lambda_1)w + \lambda_1 w^1 + \sum_{j=2}^k \lambda_j (w^j - w^1)$ for some $\lambda_1 \in [0, 1]$ and $\lambda_j \in R$ for $j = 2, \dots, k$;
- (ii) $Cx + d = \sum_{i \in I} \mu_i a^i$ for some $\mu_i \in R_+, i \in I$.

Hence, if x lies on the path S in wF_I , then the system of linear equations

$$(1) \quad \lambda_1 C(w^1 - w) + \sum_{j=2}^k \lambda_j C(w^j - w^1) - \sum_{i \in I} \mu_i a^i = -d - Cw$$

has a solution (λ, μ) such that $\lambda_1 \in [0, 1]$, $\lambda_j \in R$ for $j = 2, \dots, k$, $\mu_i \in R_+$ for $i \in I$, and x is as in (i). A linear piece of S in wF_I can be generated by making an l.p. pivoting step in (1) with one of the μ_i 's or with λ_k . Let the $n \times n$ basis matrix D consist of the n column vectors in (1) not corresponding to the pivoting variable. When we make an l.p. pivoting step in (1) with, say the column vector p , then we first compute the vector $r = D^{-1}p$. From r we obtain the direction $\bar{\lambda}_j$ for $j = 1, \dots, k$ and $\bar{\mu}_i$ for $i \in I$ into which the variables in (1) will move when we pivot the vector p into the system (1).

The next step is to determine how far we can move from the current solution. Let (λ^o, μ^o) be the current value of (λ, μ) , i.e., $(\lambda^o, \mu^o) = -D^{-1}(Cw + d)$, disregarding the pivoting variable. Then $x^o = (1 - \lambda_1^o)w + \lambda_1^o z^o$ is the current end point of the linear piece of the path S in wF_I , where $z^o \in F_I$ is given by

$$z^o = w^1 + \sum_{j=2}^k \frac{\lambda_j^o}{\lambda_1^o} (w^j - w^1).$$

Clearly, for small enough $t \geq 0$, the point $x(t)$ lies on the path in wF_I , where

$$x(t) = (1 - \lambda_1^o - t\bar{\lambda}_1)w + (\lambda_1^o + t\bar{\lambda}_1)w^1 + \sum_{j=2}^k (\lambda_j^o + t\bar{\lambda}_j)(w^j - w^1).$$

For $t \geq 0$, let $z(t)$ be such that $x(t) = (1 - \lambda_1^o - t\bar{\lambda}_1)w + (\lambda_1^o + t\bar{\lambda}_1)z(t)$, i.e.,

$$(2) \quad z(t) = w^1 + \sum_{j=2}^k \frac{\lambda_j^o + t\bar{\lambda}_j}{\lambda_1^o + t\bar{\lambda}_1} (w^j - w^1).$$

Then the point $z(t)$ lies in F_I for small enough t , and $z(0) = z^o$. From (2) we easily derive how far t can be increased from 0 until $z(t)$ lies in the boundary of F_I . As soon as for some $t_1 > 0$ we have that $(a^h)^\top z(t_1) = b_h$ for some $h \notin I$, then $z(t_1)$ lies in the boundary of F_I . From (2) it follows that

$$t_1 = \min_{i \notin I} \frac{(b_i - (a^i)^\top z^o) \lambda_1^o}{\bar{\lambda}_1 (a^i)^\top w^1 + \sum_{j=2}^k \bar{\lambda}_j (a^i)^\top (w^j - w^1) - b_i \bar{\lambda}_1},$$

where the minimum is taken over the positive denominators. We allow t_1 to be equal to plus infinity. Now let t_2 be such that the first $\mu_i = \mu_i^o + t_2 \bar{\mu}_i$ becomes zero when we increase t from 0, i.e.,

$$t_2 = \min_{\bar{\mu}_i < 0} \left\{ -\frac{\mu_i^o}{\bar{\mu}_i} \right\}.$$

When $\bar{\mu}_i \geq 0$ for all $i \in I$ we set t_2 equal to plus infinity. Let $t^* = \min\{t_1, t_2\}$ and $\lambda_1^* = \lambda_1^o + t^* \bar{\lambda}_1$. First suppose that $\lambda_1^* \geq 1$. Then for $t = (1 - \lambda_1^o) / \bar{\lambda}_1$ we obtain a

stationary point x^* on $K \cap H^-$, where

$$x^* = w^1 + \sum_{j=2}^k \left(\lambda_j^o + \frac{1 - \lambda_1^o}{\bar{\lambda}_1} \bar{\lambda}_j \right) (w^j - w^1).$$

If $0 \notin I$, then x^* is a solution of the LSPP on K . If $0 \in I$, then the next linear piece of the path S lies in F_I^+ . We will discuss how to trace such a piece in subsection 3.2.

Secondly, suppose that $\lambda_1^* < 1$ and $t_2 = t^*$, i.e., $\mu_{i'}$ becomes first zero for some $i' \in I$. If $w \in F_{I \setminus \{i'\}}$ and $0 \notin I \setminus \{i'\}$ then

$$x^* = (1 - \lambda_1^o - t^* \bar{\lambda}_1) w + (\lambda_1^o + t^* \bar{\lambda}_1) w^1 + \sum_{j=2}^k (\lambda_j^o + t^* \bar{\lambda}_j) (w^j - w^1)$$

is a solution of the LSPP on K . Otherwise, we delete i' from I , set k equal to $k + 1$, and determine a point w^k for which $(a^{i'})^\top w^k < b_{i'}$ for $i = i'$ and $(a^i)^\top w^k = b_i$ for $i \in I$. This can be done by taking w^k equal to $w^k = w^1 - (D^\top)^{-1} e(\tilde{h})$ with \tilde{h} being the index of the row $(a^{i'})^\top$ in the matrix D^\top . Now the algorithm continues by first pivoting in the vector p to replace the vector $a^{i'}$ in the matrix D^{-1} . Then a new l.p. pivoting step is made with the column vector $C(w^k - w^1)$.

Finally, suppose that $\lambda_1^* < 1$ and $t_1 = t^*$, then the point

$$z(t^*) = w^1 + \sum_{j=2}^k \frac{\lambda_j^o + t^* \bar{\lambda}_j}{\lambda_1^*} (w^j - w^1)$$

lies in the boundary of F_I . More precisely $(a^h)^\top z(t^*) = b_h$ for some $h \notin I$. We add h to I , set k equal to $k + 1$, and we move all the w^j 's parallel to $z(t^*) - z^o$ onto the affine hull of the new F_I , i.e., for $j = 1, \dots, k$, w^j becomes equal to $\hat{w}^j = w^j + \delta_j(z(t^*) - z^o)$, where

$$\delta_j = \frac{b_h - (a^h)^\top w^j}{b_h - (a^h)^\top z^o}.$$

Without loss of generality (see Kremers and Talman 1991), the points $\hat{w}^1, \dots, \hat{w}^k$ are affinely independent. We now first have to make pivoting steps in D^{-1} to replace the column vector $C(w^1 - w)$ by $C(\hat{w}^1 - w)$, $C(w^j - w^1)$ by $C(\hat{w}^j - \hat{w}^1)$ for $j = 2, \dots, k$, and, if necessary, $C(w^{k+1} - w^1)$ by the vector p . The new basis inverse matrix can be obtained by making two pivoting steps as follows.

Suppose $p = C(w^{k+1} - w^1)$. We first make a pivoting step by the vector $C(z(t^*) - z^o)$ to replace the vector $C(w^k - w^1)$ in the matrix D^{-1} . Note that $z(t^*) - z^o$ and $w^j - w^1$, $j = 2, \dots, k$, are linearly independent vectors. Otherwise, $\hat{w}^1, \dots, \hat{w}^k$ will be affinely dependent points in $\text{Aff}(F_I)$. Denote

$$D_1 = [C(w^1 - w) C(w^2 - w^1) \cdots C(w^{k-1} - w^1) C(z(t^*) - z^o) \mid a^i \mid i \in I],$$

$$D_1^{-1} = [d^1 \ d^2 \ \cdots \ d^{k-1} \ d^k \ \cdots]^\top$$

and

$$D_2 = [C(\hat{w}^1 - w) C(\hat{w}^2 - \hat{w}^1) \cdots C(\hat{w}^{k-1} - \hat{w}^1) C(z(t^*) - z^o) \cdots].$$

Since

$$C(\hat{w}^1 - w) = C(w^1 - w) + \delta_1 C(z(t^*) - z^o)$$

and

$$C(\hat{w}^j - \hat{w}^1) = C(w^j - w^1) + (\delta_j - \delta_1)C(z(t^*) - z^o) \quad \text{for } j = 2, \dots, k,$$

we have

$$D_2^{-1} = \begin{bmatrix} d^1 & d^2 & \dots & d^{k-1} & d^k - \delta_1 d^1 - \sum_{j=2}^{k-1} (\delta_j - \delta_1) d^j & \dots \end{bmatrix}^T.$$

Then we make another pivoting step by the vector $C(\hat{w}^k - \hat{w}^1)$ to replace the vector $C(z(t^*) - z^o)$ in D_2^{-1} and obtain the new inverse matrix D^{-1} .

When $p \neq C(w^{k+1} - w^1)$, i.e., $p = a^h$ for some $h \notin I$, the updating of D^{-1} is similar to the above except that at the first pivoting step the replaced vector is $C(w^{k+1} - w^1)$ and at the second pivoting step the pivoting vector is the vector p . We have shown that in both cases we need only two pivoting steps to update the matrix D^{-1} . Now we continue to trace the next linear piece of the path S by making an l.p. pivoting step with the column vector a^h .

3.2. Tracing the path in F_I^+ . Suppose that an end point x^* of a linear piece of the path S in wF_I lies in F_I for some I with $|I| = n - k + 1$ and is a solution of the LSPP on $K \cap H^-$, but not a solution of the LSPP on K . Then I must contain 0 and the next linear piece of the path S lies in F_I^+ with $\dim(F_I^+) = k$. Let the point w^o be given by $w^o = w^1 + (D^T)^{-1}e(\tilde{h})$ with \tilde{h} the row index corresponding to $(a^o)^T$ in the transpose of the current basis matrix D . Then a point x on the path S in F_I^+ can be written as

$$(i) \quad x = w^1 + (\lambda_1 - 1)(w^1 - w^o) + \sum_{j=2}^k \lambda_j (w^j - w^1);$$

$$(ii) \quad Cx + d = \sum_{i \in I} \mu_i a^i \text{ for some } \mu_i \in R_+, i \in I,$$

where $\lambda_1 \leq 1$ and $\lambda_j \in R$ for $j = 2, \dots, k$, and where w^1, \dots, w^k are affinely independent points in $\text{Aff}(F_I)$ and w^o, w^1, \dots, w^k affinely independent points in $\text{Aff}(F_I^+)$. It should be noted that the points w^1, \dots, w^k lie on the hyperplane H^o and w^o lies in the interior of the halfspace H^+ . We next change $-d - Cw$ into $-d - Cw^o$ in the right-hand side of the system (1) and make an l.p. pivoting step with the column vector $C(w^1 - w^o)$ in the new system by decreasing λ_1 from 1. In general, if x lies on the path S in F_I^+ , then the system of linear equations

$$(3) \quad \lambda_1 C(w^1 - w^o) + \sum_{j=2}^k \lambda_j C(w^j - w^1) - \sum_{i \in I} \mu_i a^i = -d - Cw^o$$

has a solution (λ, μ) such that $\lambda_1 \leq 1$, $\lambda_j \in R$ for $j = 2, \dots, k$, $\mu_i \in R_+$ for $i \in I$, and x as in (i). Therefore a linear piece of S in F_I^+ can be generated by making an l.p. pivoting step in (3) with λ_k or one of the μ_i 's. The w^j 's are such that they represent F_I^+ in the way described before. To describe an l.p. pivoting step, let D be the basis matrix for system (3) without the pivoting vector and let p be the pivoting vector. Let λ_j^o for $j = 1, \dots, k$ and μ_i^o for $i \in I$ be the current values of the variables. From $r = D^{-1}p$ we obtain the directions $\bar{\lambda}_j$ for $j = 1, \dots, k$, and $\bar{\mu}_i$ for $i \in I$ into which the variables will move when we pivot the vector p into system (3). Now let t_2

be such that for $t = t_2$ the first $\mu_i = \mu_i^o + t\bar{\mu}_i$ becomes zero when we increase t from 0, i.e.,

$$t_2 = \min_{\bar{\mu}_i < 0} \left\{ -\frac{\mu_i^o}{\bar{\mu}_i} \right\}.$$

When $\bar{\mu}_i \geq 0$ for all $i \in I$, we set t_2 equal to plus infinity. Let $x(t)$ be defined by

$$(4) \quad x(t) = w^1 + (\lambda_1^o + t\bar{\lambda}_1 - 1)(w^1 - w^o) + \sum_{j=2}^k (\lambda_j^o + t\bar{\lambda}_j)(w^j - w^1).$$

Then the point $x(t)$ lies in F_I^+ for small enough $t \geq 0$. From (4) we easily derive how far t can be increased from 0 until $x(t)$ lies on the boundary of F_I^+ . As soon as for some $t_1 > 0$ we have $(a^h)^\top x(t_1) = b_h$ for some $h \notin I$, then the point $x(t_1)$ lies in the boundary of F_I^+ . From (4) it follows that

$$t_1 = \min_{i \notin I} \frac{b_i - (a^i)^\top x(0)}{\sum_{j=2}^k \bar{\lambda}_j (a^i)^\top (w^j - w^1) - \bar{\lambda}_1 (a^i)^\top (w^1 - w^o)},$$

where the minimum is taken over the positive denominators. We allow t_1 to be equal to plus infinity. Let $t^* = \min\{t_1, t_2\}$ and $\lambda_1^* = \lambda_1^o + t^*\bar{\lambda}_1$.

Suppose $t_2 = t^*$ and $\lambda_1^* < 1$, i.e., $\mu_{i'}$ becomes first zero for some $i' \in I$. When $i' = 0$ then the point

$$x^* = w^1 + (\lambda_1^o + t^*\bar{\lambda}_1 - 1)(w^1 - w^o) + \sum_{j=2}^k (\lambda_j^o + t^*\bar{\lambda}_j)(w^j - w^1)$$

solves the LSPP on K . When $i' \neq 0$ we update the set I by deleting i' from I and set k equal to $k + 1$. We have to determine a point w^k for which $(a^i)^\top w^k < b_i$ for $i = i'$ and $(a^i)^\top w^k = b_i$ for $i \in I$. In order to do this we take the point $w^k = w^1 - (D^\top)^{-1}e(\tilde{h})$, where \tilde{h} corresponds to the index of the row $(a^{i'})^\top$ in the matrix D^\top . Note that the new point w^k also lies on the hyperplane H^o because $(a^o)^\top$ is a row vector of D^\top . Now the algorithm continues by first pivoting in the column vector p to replace the vector $a^{i'}$ in the matrix D^{-1} . Then a new l.p. pivoting step is made with the column vector $C(w^k - w^1)$ to trace the next linear piece of the path S .

Next suppose that $t_1 = t^*$ and $\lambda_1^* < 1$; then the point $x(t^*)$ lies in the boundary of F_I^+ , i.e., $(a^h)^\top x(t^*) = b_h$ for some $h \notin I$. We add h to I and set k equal to $k - 1$. Let ν_o be such that the point $\tilde{y}^o = (1 - \nu_o)w + \nu_o x(0)$ lies in H^o and let ν_1 be such that the point $\tilde{y}^1 = (1 - \nu_1)w + \nu_1 x(t^*)$ lies in H^o , i.e.,

$$\nu_o = \frac{b_o - (a^o)^\top w}{(a^o)^\top (x(0) - w)} \quad \text{and} \quad \nu_1 = \frac{b_o - (a^o)^\top w}{(a^o)^\top (x(t^*) - w)}.$$

Then we move w^o, w^1, \dots, w^k parallel to the direction $\tilde{y}^1 - \tilde{y}^o$ until they lie on the hyperplane $(a^h)^\top x = b_h$, i.e., w^j becomes equal to

$$\hat{w}^j = w^j + \frac{b_h - (a^h)^\top w^j}{(a^h)^\top (\tilde{y}^1 - \tilde{y}^o)} (\tilde{y}^1 - \tilde{y}^o) \quad \text{for } j = 0, 1, \dots, k.$$

Now update D by replacing $C(w^1 - w^o)$ by $C(\hat{w}^1 - \hat{w}^o)$, $C(w^j - w^1)$ by $C(\hat{w}^j - \hat{w}^1)$ for $j = 2, \dots, k$, and if necessary $C(w^{k+1} - w^1)$ by p . The updating of D^{-1} is similar to the description in subsection 3.1 except for the use of $C(\bar{y}^1 - \bar{y}^o)$ instead of $C(z(t^*) - z^o)$. The algorithm continues by pivoting into the system the column a^h .

Finally if $\lambda_1^* \geq 1$, then for $t^* = (1 - \lambda_1^o)/\bar{\lambda}_1$ the point x^* given by

$$x^* = w^1 + \sum_{j=2}^k (\lambda_j^o + t^* \bar{\lambda}_j) (w^j - w^1)$$

is a solution of the LSPP on $K \cap H^-$. The next linear piece of the path S lies in wF_I . We replace $-d - Cw^o$ by $-d - Cw$ in the right-hand side of the system (3) and make an l.p. pivoting step with the column vector $C(w^1 - w)$ in this new system by decreasing λ_1 from 1. Note that we return to the system (1) again.

The path S leading from the arbitrary point w to a solution to the LSPP on K consists of a finite number of linear pieces. Each piece lies in wF_I or in F_I^+ for some index set I and can be followed by making just one linear programming pivoting step in the way described above. Under the condition stated at the end of §2 the algorithm solves the problem within a finite number of steps.

REMARK 1. If K is not simple, then at a solution (λ^o, μ^o) of (1) or (3) the set $I = \{i | \mu_i^o > 0\}$ might be a proper subset of the set G defined by $G = \{i | (a^i)^\top x = b_i \text{ for all } x \in F_I\}$ in case of (1) or $G = \{0\} \cup \{i | (a^i)^\top x = b_i \text{ for all } x \in F_I^+\}$ in case of (3). If in such a case after a ratio test $\mu_{i'}$ becomes 0 for some $i' \in I$ and the end point x is not a solution, then we should first check whether some $\mu_{j'}$, $j' \in G \setminus I$, can be increased from 0 to extend the linear piece of the path. If so, then a pivoting step is made with $a^{j'}$ and the algorithm continues as described above. If not, the algorithm continues as above in the set $wF_{I \setminus \{i'\}}$ or in $F_{I \setminus \{i'\}}^+$ having one dimension higher.

REMARK 2. The homotopy behind the algorithm can be interpreted by the concept of an expanding set (see van der Laan and Talman 1986 and Yamamoto 1987). This set expands from the starting point to the set K and the algorithm traces a path of stationary points on the set as it expands. For $t \geq 0$ let

$$K(t) = \{x | x = (1 - t)^+ w + (1 + (t - 1)^-) z$$

$$\text{for } z \in K, (a^o)^\top z \leq b_o + (t - 1)^+ b_o\},$$

where $\alpha^+ = \max\{0, \alpha\}$ and $\alpha^- = \min\{0, \alpha\}$ for $\alpha \in R$. For $t \geq 0$, let us consider the stationary point problem on $K(t)$ with respect to the function f . Then we have

- (a) $x = w$ is the unique stationary point of f on $K(0)$;
- (b) if x is a stationary point of f on $K(t)$ for some $t < 1$, then $x = (1 - t)w + tz$ for some $z \in F_I$ with I such that $f(x) \in N(F_I)$;
- (c) if x is a stationary point of f on $K(t)$ for some $t \geq 1$, then $x \in F_I^+$ and $f(x) \in N(F_I)$ for some I .

By tracing the solution path of the stationary point problem on $K(t)$ for $t \geq 0$ from $(z, t) = (w, 0)$ we therefore obtain the path generated by the algorithm.

Acknowledgement. The authors would like to thank one of the referees for his helpful comments. Part of this research was done when the first author was visiting the Center for Economic Research (CentER), Tilburg University, Tilburg, The Netherlands. The author is grateful to CentER for the invitation and hospitality.

References

- [1] Dai, Y., van der Laan, G., Talman, A. J. J., and Yamamoto, Y. (1991). A Simplicial Algorithm for the Nonlinear Stationary Point Problem on an Unbounded Polyhedron. *SIAM J. Control Optim.* **1** 151-165.
- [2] Kamiya, K. and Talman, A. J. J. (1990). *Linear Stationary Point Problems*. Discussion Paper No. 9022, Center for Economic Research, Tilburg University, Tilburg, The Netherlands.
- [3] Kremers, J. A. W. M. and Talman, A. J. J. (1991). *A SLSP-algorithm to Compute an Equilibrium in an Economy with Linear Production Technologies*, FEW 496, Department of Econometrics, Tilburg University, Tilburg, The Netherlands.
- [4] van der Laan, G. and Talman, A. J. J. (1986). On the Computation of Fixed Points in the Product Space of Unit Simplices and an Application to Noncooperative N -person Games. *J. Optim. Theory Appl.* **50** 262-281.
- [5] Yamamoto, Y. (1987). A Path Following Algorithm for Stationary Point Problems. *J. Oper. Res. Soc. Japan* **30** 181-198.

Y. Dai: Department of Management Science, Kobe University of Commerce, Nishi-ku, Kobe 651-21, Japan

D. Talman: Department of Econometrics, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands